Weak-shot Semantic Segmentation via Dual Similarity Transfer

Anonymous Author(s)
Affiliation
Address
email

Abstract

Semantic segmentation is a practical and active task, but severely suffers from the expensive cost of pixel-level labels when extending to more classes in wider applications. To this end, we focus on the problem named weak-shot semantic segmentation, where the novel classes are learnt from cheaper image-level labels with the support of base classes having off-the-shelf pixel-level labels. To tackle this problem, we propose a dual similarity transfer framework, which is built upon MaskFormer to disentangle the semantic segmentation task into single-label classification and binary segmentation for each proposal. Specifically, the binary segmentation sub-task allows proposal-pixel similarity transfer from base classes to novel classes, which enables the mask learning of novel classes. We also learn pixel-pixel similarity from base classes and distill such class-agnostic semantic similarity to the semantic masks of novel classes, which regularizes the segmentation model with pixel-level semantic relationship across images. In addition, we propose a complementary loss to facilitate the learning of novel classes. Comprehensive experiments on the challenging COCO-Stuff-10K and ADE20K datasets demonstrate the effectiveness of our method.

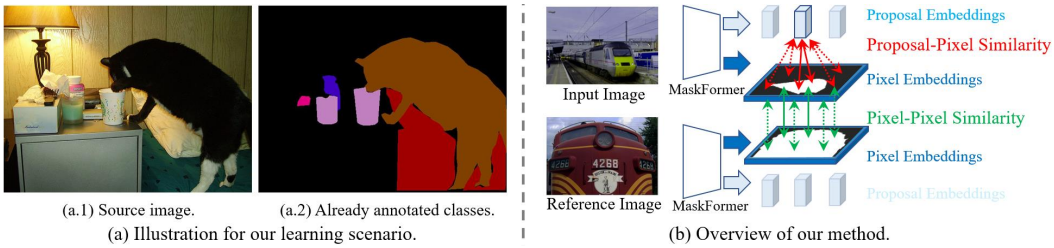
1 Introduction

Semantic segmentation [26, 5, 40, 42] is a fundamental and active task in computer vision, which aims to produce class label for each pixel in image. Training modern semantic segmentation models usually requires pixel-level labels for each class in each image, and thus costly datasets have been constructed for learning. As a practical task, there is a growing requirement to segment more classes in wider applications. However, annotating dense pixel-level mask is too expensive to cover the continuously increasing classes, which dramatically limits the applications of semantic segmentation.

In practice, we have some already annotated semantic segmentation datasets, *e.g.*, COCO-80 [24]. These costly datasets focus on certain classes (*e.g.*, the 80 classes in [24]), and ignore other uninterested classes (*e.g.*, other classes beyond the 80 classes in [24]). Over time, there may be demand to segment more classes, *e.g.*, extending to COCO-171 [2]. We refer to the off-the-shelf classes having already annotated masks as base classes, and those newly covered classes as novel classes. As illustrated in Fig. 1 (a), one sample in COCO-80 focuses on segmenting *cat*, *bed*, etc, but ignores the *lamp*. Existing scenario for expansion in [2] is to annotate the dense pixel-level masks for novel classes again (*i.e.*, annotating the mask for *lamp* in Fig. 1 (a)), which is too expensive to scale-up. To this end, we focus on a cheaper yet effective scenario in this paper, where we only need to annotate image-level labels for the novel classes. That is, we just tag the source image in Fig. 1 (a) with class label *lamp*, instead of annotating pixel-level mask as in [2].

We refer to our learning scenario as weak-shot semantic segmentation, which focuses on further segmenting novel classes by virtue of cheaper image-level labels with the support of base classes

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a.1) Source image. (a.2) Already annotated classes.

(a) Illustration for our learning scenario. (b) Overview of our method.

Figure 1: As shown in (a), off-the-shelf datasets focus on certain base classes, *i.e.*, the colored masks in (a.2), while leaving the novel classes ignored, *i.e.*, the regions in black. We would like to segment novel classes by virtue of cheaper image-level labels. As shown in (b), our method obtains proposal embeddings and pixel embeddings for each image based on MaskFormer [8]. Then, the transferred proposal-pixel similarity could produce masks for proposal embeddings of novel classes, and the transferred pixel-pixel similarity could provide semantic relation regularization for pixel pairs across images. The solid (*resp.*, dotted) arrow indicates similar (*resp.*, dissimilar). Best viewed in colors.

having pixel-level masks. Specifically, given a standard semantic segmentation dataset annotated only for base classes (the novel classes hide in the ignore regions). We assume that we have the image-level labels about novel classes in each image, we would like to segment all the base classes and novel classes in the test stage. Similar scenario has been explored in RETAB [46], but they 1) further assume that the off-the-shelf dataset contains no novel classes; 2) assume that the *background* class (in PASCAL VOC dataset [10]) consisting of all ignore classes is annotated with pixel-level mask. However, these two assumptions are hard to satisfy in real-world applications, because the ignore region of off-the-shelf dataset may contain novel classes and we actually cannot have the masks of *background* class before having the masks of novel classes. In contrast, our scenario is similar in spirit but more succinct and practical, which is well demonstrated by the aforementioned instance expanding from COCO-80 to COCO-171. Besides, RETAB [46] only conducted experiments on PASCAL VOC, while we focus on more challenging datasets (*i.e.*, COCO-Stuff-10K [2] and ADE20K [45]), without a *background* class annotated with masks.

In weak-shot semantic segmentation, the key problem is how to learn dense pixel-level masks from the image-level labels of novel classes with the support of pixel-level masks of base classes. Our proposed solution is dual similarity transfer, *i.e.*, proposal-pixel similarity transfer and pixel-pixel similarity transfer, which is built upon MaskFormer [8] as shown in Fig. 1 (b). We choose MaskFormer [8] because it disentangles the segmentation task into two sub-tasks: proposal classification and proposal segmentation. Specifically, MaskFormer [8] produces some proposal embeddings (*aka* per-segment embeddings in [8]) from shared query embeddings for input image, each of which is assigned to be responsible for one class present in image (allowing empty assignment). Then MaskFormer performs proposal classification sub-task and proposal segmentation sub-task for each proposal embedding according to their assigned classes respectively.

In our setting and framework, the proposal embeddings of base classes are supervised in both sub-tasks using class and mask annotations, while the proposal embeddings of novel classes are supervised only in classification sub-task due to lacking mask annotations. In this way, the proposal embeddings of novel class could still produce effective masks by proposal-pixel similarity transferred from base classes. Because the binary segmentation sub-task of base classes is learning proposal-pixel similarity, which is class-invariant. Nevertheless, without any supervision, the binary masks of novel classes are relatively insufficient to be consistent with the semantic class in dataset. Thus we further propose pixel-pixel similarity transfer, which learns the pixel pair-wise semantic similarity from base classes and distills such class-invariant similarity into the produced masks of novel classes. Therefore, the model is supervised to produce segmentation results containing semantic relationship for novel classes. Inspired by [34, 37, 33], we learn from and distill to cross-image pixel pairs, which could also introduce global context into model, *i.e.*, enhancing the pixel semantic consistency across images. In addition, we propose a complementary loss, based on the insight that the union set of pixels belonging to base classes is complementary to the union set of pixels belonging to novel classes or ignore class in each image, which provides supervision for the union of masks of novel classes.

We conduct extensive experiments on two challenging datasets (*i.e.*, COCO-Stuff-10K [2] and ADE20K [45]) to demonstrate the effectiveness of our method. We summarize our contributions as

1) We propose a dual similarity transfer framework for weak-shot semantic segmentation based on MaskFormer, which is effective for proposal-pixel similarity transfer.

2) We propose pixel-pixel similarity transfer, which learns pixel-pixel semantic similarity from base classes and distills such class-agnostic similarity to the segmentation results of novel classes. We also propose a complementary loss to facilitate the mask learning of novel classes.

3) Extensive experiments on the challenging COCO-Stuff-10K [2] and ADE20K [45] datasets demonstrate the practicality of our scenario and the effectiveness of our method.

2 Related Works

Weakly-supervised Semantic Segmentation (WSSS). Considering the expensive cost for annotating pixel-level masks, WSSS [29, 11, 31, 19] rely on only image-level labels to train the segmentation model, which has attracted increasing attention. The major WSSS methods [3, 1, 36] firstly train a classifier to obtain class activation map (CAM) [44] to derive pseudo masks, which are then used to train a standard segmentation model. For example, SEC [17] proposed "seed, expand, and constrain" three principles, which produced a profound impact to WSSS. Under similar pipeline, some works [21, 3, 36] focus on enhancing the seed, while some other works [16, 35, 41] pay attention to improving the expand strategy. Although WSSS has achieved great success, the expanded CAM is difficult to cover the full semantic class due to lacking annotation about the spatial scope of class. Fortunately, in our focused problem, such information could be derived from an off-the-shelf dataset and transferred to facilitate the learning of classes with only image-level labels.

Weak-shot Learning. Reducing the annotation cost is a practical and extensive demand for various applications of deep learning. Recently, weak-shot learning focusing on learning novel classes from cheap weak labels with the support of a set of base classes having off-the-shelf strong labels has been explored in image classification [4], object detection [25, 43, 7, 22], and instance segmentation [18], which has achieved promising success. In weak-shot semantic segmentation [46], the problem is learning to segment novel classes with only image-level labels with the support of base classes having pixel-level labels. Concerning the task setting, as aforementioned, RETAB [46] further assumes that the off-the-shelf dataset has no novel classes and the *background* class is annotated with pixel-level mask, while the setting in this paper is more succinct and practical. Concerning the technical method, RETAB [46] follows the framework of WSSS, which suffers from a complex and inconvenient multi-stage pipeline, *i.e.*, training classifier, deriving CAM, expanding to pseudo-labels, and re-training. In contrast, we build our framework based on MaskFormer [8] to perform dual similarity transfer, which could achieve satisfactory performance in single-stage without re-training.

Similarity Transfer. As an effective method, similarity transfer has been widely applied in various transfer learning tasks [6, 32, 4]. Specifically, semantic similarity (whether the two inputs belong to the same class) is class-agnostic, and thus is transferrable across classes. To name a few, Guo *et al*. [13] transferred class-class similarity and sample-sample similarity across domains in active learning. CCN [15] proposed to learn semantic similarity between image pair, which is robust in both cross-domain and cross-task transfer learning. PrototypeNet [30] proposed to learn and transfer the similarity between image and prototype across categories for both few-shot classification and zero-shot classification. In this paper, we propose to transfer proposal-pixel similarity and pixel-pixel similarity for weak-shot semantic segmentation. The two kinds of similarities both belong to semantic similarity, which are highly transferrable across classes.

3 Methodology

3.1 Problem Definition

In our weak-shot semantic segmentation, we are given a standard segmentation dataset annotated for base classes $\mathcal{C}_\mathrm{b}$, and we would like to further segment another set of novel classes $\mathcal{C}_\mathrm{n}$ ignored in the off-the-shelf dataset, where $\mathcal{C}_\mathrm{b} \cap \mathcal{C}_\mathrm{n} = \emptyset$. We assume that we have the image-level labels for $\mathcal{C}_\mathrm{n}$, which is rather cheaper and more convenient to obtain than pixel-level mask. In summary, for each training image, we have image-level labels for both $\mathcal{C}_\mathrm{b}$ and $\mathcal{C}_\mathrm{n}$, and we only have pixel-level masks for $\mathcal{C}_\mathrm{b}$. In the test stage, we need to predict pixel-level masks for both $\mathcal{C}_\mathrm{b}$ and $\mathcal{C}_\mathrm{n}$.

3.2 Review of MaskFormer

In this section, we briefly review MaskFormer [8], which lays the foundation of our framework. The general pipeline of MaskFormer disentangles semantic segmentation task into two sub-tasks: proposal classification and proposal segmentation. Specifically, MaskFormer maintains N learnable query embeddings $\mathcal{Q} \in \mathbb{R}^{C\times N}$ shared for all images, as shown in the upper part of Fig. 2. When the image is input, the backbone features are extracted via a backbone. Then, N query embeddings attend to backbone features to produce proposal embeddings $\mathcal{E}_{\text{prop}} \in \mathbb{R}^{C\times N}$ via a transformer decoder. For each proposal embedding, a bipartite matching algorithm assigns a class present in the input image to it, considering the classification loss and mask loss as the cost for assignment. For proposal classification sub-task, each proposal embedding is fed to a linear classifier to yield class probability predictions $\mathcal{Y} \in \mathbb{R}^{(K+1)\times N}$ over K semantic classes and 1 ignore class. For proposal segmentation sub-task, pixel embeddings $\mathcal{E}_{\text{pix}} \in \mathbb{R}^{C\times H\times W}$ are extracted from backbone features via a pixel decoder. Afterwards, proposal embeddings are processed by several FC layers and computes dot-product followed by sigmoid with pixel embeddings to produce binary masks $\mathcal{M} \in \mathbb{R}^{N\times H\times W}$, *i.e.*, $\mathcal{M}[i,h,w] = \text{sigmoid}(\mathcal{E}_{\text{prop}}[:,i] \cdot \mathcal{E}_{\text{pix}}[:,h,w])$ (processing FCs are omitted for brevity). In the training stage, the two sub-tasks for each proposal embedding are supervised towards to the label and mask of assigned class (the mask loss of ignore class is eliminated). In the test stage, the semantic segmentation result at pixel (h,w) is obtained via marginalization over probability-mask pairs, *i.e.*, $\arg\max_{c\in\{1,\ldots,K\}} \sum_{i=1}^{N} \mathcal{Y}[c,i] \cdot \mathcal{M}[i,h,w]$. For more details, please refer to MaskFormer [8].

3.3 Proposal-Pixel Similarity Transfer on MaskFormer

In our setting, we have only image-level labels for novel classes and we choose to produce pixel-level masks via proposal-pixel similarity transfer based on MaskFormer [8]. For concise description, we refer to the proposal embeddings assigned with base (*resp.*, novel or ignore) classes as base (*resp.*, novel or ignore) proposal embeddings. We refer to the binary (*resp.*, novel or ignore) masks produced from base proposal embeddings as base (*resp.*, novel or ignore) masks. We refer to the image pixels actually belonging to base (*resp.*, novel or ignore) classes as base (*resp.*, novel or ignore) pixels.

Our framework is illustrated in Fig. 2, where the upper part is the same as MaskFormer producing proposal embeddings $\mathcal{E}_{prop} \in \mathbb{R}^{C\times N}$. Because we have no mask for novel classes in the bipartite matching algorithm, we omit the mask loss in the cost if the class to be assigned belongs to $\mathcal{C}_{\text{n}}$. For the proposal classification sub-task, all proposal embeddings are supervised to predict the assigned classes because we have all the image-level labels of each input image. Thus, we have the classification loss

$$\mathcal{L}_{\text{cls}} = \sum_{i=1}^{N} -\log(\mathcal{Y}[y^*(i), i]), \tag{1}$$

where $y^*(i)$ indicates the assigned single ground-truth (GT) class of the i-th proposal embedding. For the binary segmentation sub-task, only the base masks are supervised as

$$\mathcal{L}_{\text{mask}} = \sum_{y^*(i)\in\mathcal{C}_{\text{b}}} \mathcal{L}_{\text{focal+dice}}(\mathcal{M}[i,:,:], M^*(y^*(i))), \tag{2}$$

where $M^*(y^*(i))$ indicates the GT mask of the assigned class of the i-th proposal, and $\mathcal{L}_{\text{focal+dice}}$ is a binary mask loss consisting of a focal loss [23] and a dice loss [27] as in MaskFormer [8]. Under the supervision of mask loss, the base proposal embeddings will produce a binary similarity with all the pixel embeddings, *i.e.*, each proposal embedding is *similar* or *dissimilar* to each pixel embedding. Such binary similarity is class-agnostic, and thus could transfer across classes. Therefore, although the novel proposal embeddings are not supervised to predict GT masks, they could rely on the transferred similarity for producing segmentation masks.

3.4 Pixel-Pixel Similarity Transfer via Pixel SimNet

Due to lacking semantic supervision of novel classes, the novel masks are hard to precisely reach a semantic consistency across training images, so we design a novel strategy to learn the pair-wise semantic similarity from base pixels across images and distill such class-agnostic similarity to novel pixels across images, which could complement to the above proposal-pixel similarity.

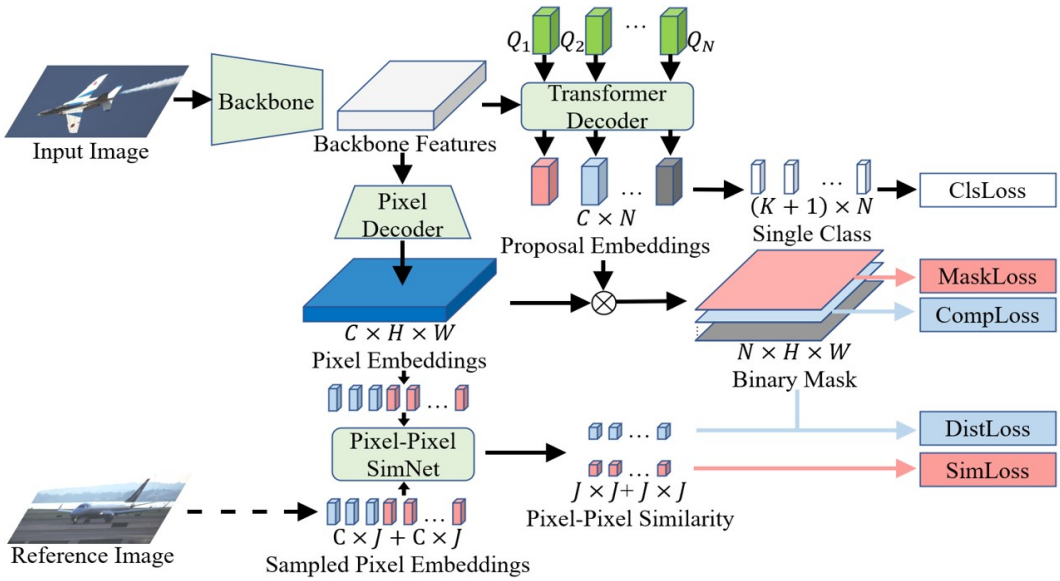


Figure 2: The detailed illustration of our framework. As in MaskFormer, we produce N proposal embeddings in each image. On the one hand, each proposal embedding is fed to the linear classifier, where both base and novel classes are supervised by classification loss (ClsLoss). On the other hand, the similarities between each proposal embedding and pixel embeddings are computed to produce binary mask, where only base masks (in red) are supervised by GT mask (MaskLoss) while novel masks (in blue) are supervised by complementary loss (CompLoss). We sample some pixels and construct pixel pairs across two images. The concatenated pixel embeddings are fed to SimNet, where the base pixel pairs (in red) are used to train SimNet with similarity loss (SimLoss) and novel pixel pairs (in blue) are used for similarity distillation (DistLoss).

Specifically, to learn pixel-pixel similarity across images, for each input image, we randomly sample another training image as its reference image. According to image-level labels, we ensure that the input image and reference image have common base and novel classes. We train a pixel-pixel similarity net (SimNet) using base pixels which are associated with GT pixel-level labels. By class-aware random sampling J base pixels from both images, we construct $J \times J$ ($J = 100$ in our experiments) pixel pairs balanced in binary semantic class (*i.e.*, *similar v.s. dissimilar*). We concatenate the pixel embeddings of paired base pixels to obtain pixel-pair embeddings $\mathcal{E}_{\mathrm{pair-b}} \in \mathbb{R}^{(C+C)\times J\times J}$. A SimNet (3 FC layers followed by sigmoid function) takes in $\mathcal{E}_{\mathrm{pair-b}}$ and outputs the similar score $\mathcal{R}_{\mathrm{b}} \in \mathbb{R}^{J\times J}$ for the base pixel pairs. We have GT labels for the pixels from base region, and thus $\mathcal{R}_{\mathrm{b}}$ is supervised by simple binary classification loss $\mathcal{L}_{\mathrm{sim}}$ for training SimNet.

To distill the learned pixel-pixel semantic similarity into the segmentation of novel pixels, we also construct $J \times J$ pixel pairs from the not-base region (ignore pixels are rather fewer relatively to novel pixels) of both images respectively and obtain $\mathcal{E}_{\mathrm{pair-n}} \in \mathbb{R}^{(C+C)\times J\times J}$. Because the input image and reference image have common novel classes, the sampled pixel pairs have chance to belong to the same novel classes, which could inject semantic consistency (*i.e.*, *similar* class) into novel pixels. We use the SimNet to estimate the GT semantic similarity and obtain $\mathcal{R}_{\mathrm{n}} \in \mathbb{R}^{J\times J}$, which is detached of gradient and employed as the distillation source. For the distillation target, we compute the segmentation scores of novel classes at sampled pixel, *i.e.*, $\mathcal{S}_{\mathrm{n}} = \{\sum_{i=1}^{N} \mathcal{Y}[c,i] \cdot \mathcal{M}[i,:,:]\}_{c\in\mathcal{C}_{\mathrm{n}}}$, and collect the scores of pixel sampled from input image $\mathcal{S}_{\mathrm{input-n}} \in \mathbb{R}^{|\mathcal{C}_{\mathrm{n}}|\times J}$ and the scores from reference image $\mathcal{S}_{\mathrm{ref-n}} \in \mathbb{R}^{|\mathcal{C}_{\mathrm{n}}|\times J}$. The pixel-pixel similarity distillation from base classes to novel classes is achieved by

$$\mathcal{L}_{\mathrm{dist}} = \mathrm{BCE}(\mathrm{Relu}(\mathrm{Cos}_{\mathrm{enum}}(\mathcal{S}_{\mathrm{input-n}}, \mathcal{S}_{\mathrm{ref-n}})), \mathcal{R}_{\mathrm{n}}), \tag{3}$$

where $\mathrm{Cos}_{\mathrm{enum}}(\cdot,\cdot)$ computes the cosine similarity of all enumerated pixel pairs between the two sets of $|\mathcal{C}_{\mathrm{n}}|$-dim vectors, $\mathrm{Relu}(\cdot)$ focuses on the non-negative part considering that zero value means independent and thus dissimilar, and $\mathrm{BCE}(\cdot,\cdot)$ is the binary cross entropy function for pushing the

distillation target towards the distillation source. In this way, the pixel-pixel similarity is learned from base classes and transferred to facilitate the semantic consistency of novel masks across images.

3.5 Complementary Loss

Although we have no pixel-level annotation for novel classes, we have the prior knowledge that the union set of novel pixels and ignore pixels and the union set of base pixels are complementary in each image. Therefore, we use a complementary loss to inject such insight into our model:

$$\mathcal{L}_{\text{comp}} = \mathcal{L}_{\text{focal+dice}}(\mathbf{1} - \bigcup_{y^*(i)\in\mathcal{C}_{\text{b}}} M^*(y^*(i)), \bigcup_{y^*(i)\in\mathcal{C}_{\text{n}}+\{\text{ignore}\}} \mathcal{M}[i,:,:]), \tag{4}$$

where the first input is the complementary set of the union set of all the GT masks of base classes, the second input is the union set of all the binary masks produced by the proposal embeddings assigned with novel classes or ignore class, and the union operator specifically takes the maximum value over involved binary masks at each pixel. Because the ignore mask produced from ignore proposal embedding is meaningless here as in MaskFormer [8], we reset a constant value γ (default 0.1) for the ignore mask, which is a prior probability for the ignore class. The constant prior probability is similar to the background threshold used in weakly supervised semantic segmentation [20, 36, 38].

3.6 Training and Inference

Our training and inference pipelines are the same as MaskFormer [8], and the full training loss is

$$\mathcal{L}_{\text{full}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{sim}} + \alpha\mathcal{L}_{\text{dist}} + \beta\mathcal{L}_{\text{comp}}, \tag{5}$$

where $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{mask}}$ implicitly include the default trade-off hyper-parameters in MaskFormer, and $\mathcal{L}_{\text{sim}}$ is actually also a classification loss and balances well with $\mathcal{L}_{\text{cls}}$. We use α (default value 0.1) for balancing the distillation loss and β (default value 0.2) for balancing the complementary loss.

4 Experiments

4.1 Datasets, Evaluation, and Implementation Details

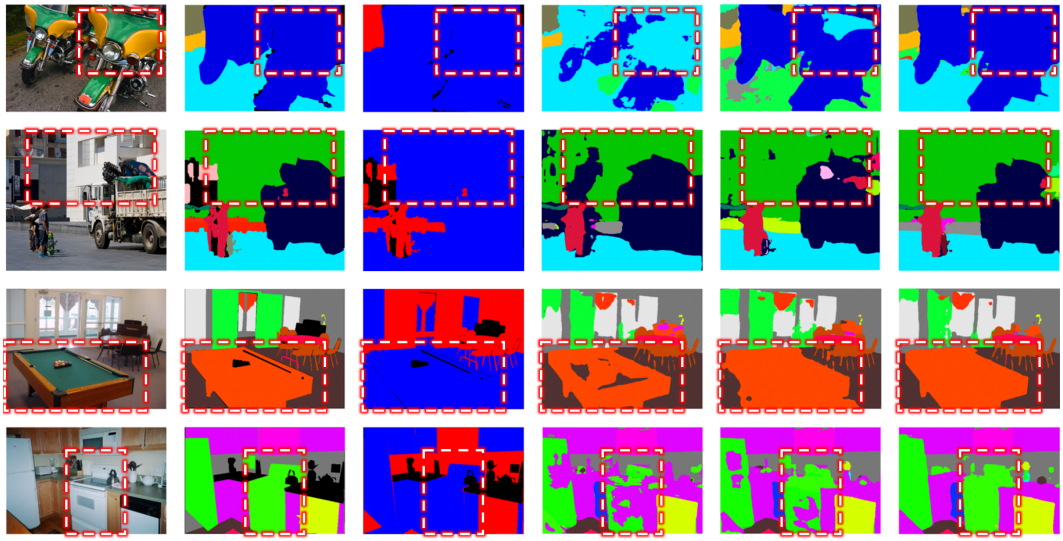
We explore the learning scenario and evaluate our method on two challenging datasets. COCO-Stuff-10K [2] contains 9k training images and 1k test images, covering 171 semantic classes. ADE20K [45] has 20k training images and 2k validating images, covering 150 semantic classes. All the data preparing processes follow those in MaskFormer [8]. For the learning scenario, we random split all the classes in the dataset into base classes and novel classes. Then we keep the image-level labels of novel classes in each training image, and reset the pixel-level masks of novel classes to ignore class (*e.g.*, class ID 255). We follow the split ratio $|\mathcal{C}_{\text{b}}| : |\mathcal{C}_{\text{n}}| = 3 : 1$ in RETAB [46], which are also widely used in few-/zero-shot learning [28, 12]. We employ Intersection-over-Union (IoU) as our evaluation metric. To show the performance more clearly, we focus on mean IoU of novel classes, because the base classes are always supervised by GT masks and show relatively robust performances. For the network architecture, we follow the setting of MaskFormer and use ResNet50 [14] pre-trained on ImageNet [9] as backbone. More details are left to Supplementary.

4.2 Comparison with Prior Works

Comparative Baselines. We set three groups of baselines: 1) representative WSSS methods, including SEAM [36], CONTA [38], CPN [39], and RIB [20]; 2) weak-shot segmentation segmentation methods, including RETAB [46]; 3) the standard segmentation method, a MaskFormer [8] trained with GT masks of novel classes, named *FullyOracle*. Because WSSS baselines and RETAB all need a re-training stage to achieve the final performance and there are GT masks of base classes in training images, we re-train all baselines with mixed labels. Specifically, the mixed labels consist of GT masks of base classes and pseudo-labels of novel classes generated by the model (*e.g.*, deriving and expanding CAM). For a fair comparison, we re-train each baseline on the standard MaskFormer [8] using the generated mixed labels. Therefore, all the methods are consistent in network architecture. More details about baseline setting are left to Supplementary.

Table 1: Comparison of various methods on two benchmark datasets in four splits. The methods marked by * are appended with a re-training stage. The best results are highlighted in boldface.

Method	COCO-Stuff-10K [2]				ADE20K [45]			
	Split1	Split2	Split3	Split4	Split1	Split2	Split3	Split4
SEAM* [36]	22.2	17.1	22.4	21.5	25.6	22.2	20.0	21.1
CONTA* [38]	23.5	17.7	23.2	22.5	26.7	23.6	20.6	21.6
CPN* [39]	23.4	18.9	23.8	23.0	27.1	24.2	20.8	21.9
RIB* [20]	24.1	18.7	24.4	23.3	28.7	25.9	21.3	22.2
RETAB* [46]	29.3	23.6	29.9	29.4	33.8	30.8	25.9	26.3
Ours	32.9	27.0	31.6	33.3	36.2	32.9	27.3	28.7
Ours*	**34.7**	**27.8**	**33.7**	**35.2**	**38.0**	**35.4**	**30.6**	**32.2**
FullyOracle	37.8	34.6	43.2	39.4	46.9	43.8	38.1	35.8



(a) Input Image (b) GT Mask (c) Base Red; Novel Blue (d) WSSS (e) RETAB (f) Ours

Figure 3: Visual comparison on COCO-Stuff-10K (the first two rows) and ADE20K (the last two rows) datasets. Columns (a,b) show input images and GT masks. Column (c) shows the base/novel split maps indicating base pixels by red and novel pixels by blue. Columns (d,e,f) show the results of three methods. The dotted rectangles highlight the major differences.

Quantitative Comparison. We summary all the results on four random splits of two datasets in Tab. 1. Even without re-training, our method can dramatically outperforms WSSS baselines and RETAB [46] (*e.g.*, about 2.6% against RETAB), demonstrating the effectiveness of our method based on dual similarity transfer. By appending a re-training stage as in WSSS, our method is further improved by about 2.2%. Compared with the fully-supervised upper bound *Fully Oracle*, our method with re-training could achieve satisfactory performances (*i.e.*, about $78\% \sim 92\%$ of the upper bound) on the two challenging datasets, showing the potential of our dual similarity transfer framework for learning novel classes from only image-level labels.

Qualitative Comparison. We conduct qualitative comparison for the re-trained WSSS baseline RIB [20], RETAB [46] and our method on both datasets. As shown in Fig. 3, WSSS baseline tends to produce incomplete masks, and our method could produce more complete and precise semantic masks against RETAB [46]. Even in complex scenes (*e.g.*, the 2nd and 3rd rows), our method still predicts preferable results for novel classes. More visualizations are left to Supplementary.

4.3 Module Analysis

We quantitatively and qualitatively investigate each module of our model on COCO-Stuff-10K dataset.

Table 2: Module contributions on four splits of COCO-Stuff-10K. "Pr" is proposal-similarity transfer, "Pi" is pixel-pixel similarity transfer, and "Co" is complementary loss.

Pr	Pi	Co	S1	S2	S3	S4
√			25.3	20.4	26.7	25.8
√	√		29.2	24.7	29.5	30.4
√		√	28.8	24.6	29.2	29.4
√	√	√	32.9	27.0	31.6	33.3

Table 3: The transferability of pixel-pixel similarity on four splits of COCO-Stuff-10K. We use F1-score as the metric for the binary classification task (*i.e.*, *Dissimilar v.s. Similar*).

		S1	S2	S3	S3
Base	*Dis*	86.7	87.8	93.9	94.7
	Sim	89.3	89.1	86.6	87.4
Novel	*Dis*	80.4	79.5	80.5	83.7
	Sim	75.6	73.1	73.4	78.5

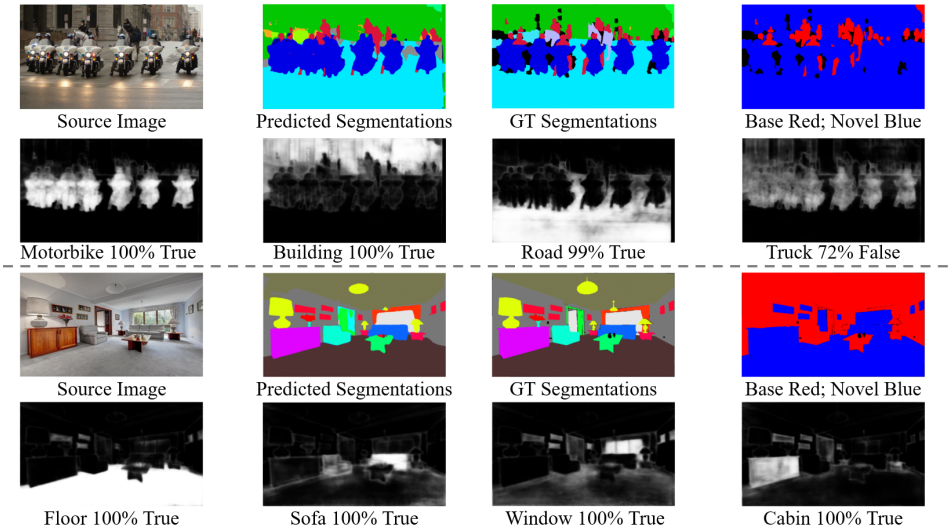


Figure 4: Visualizations for the novel proposals on COCO-Stuff-10K (upper) and ADE20K (lower) datasets. In each example, the first row shows the source image, predicted segmentation, GT segmentation, and base/novel split map, and the second row shows the first four proposals ranked by the predicted scores of novel classes. Each caption of the proposal sub-figure describes: 1) the class name; 2) the predicted class score; 3) whether the class is actually present in the image.

Proposal-pixel Similarity Transfer. As shown in the 1st row of Tab. 2, only with proposal-pixel similarity transfer, the model could achieve acceptable performance on various dataset splits, which lays the foundation of our method. We provide in-depth qualitative visualization in Fig. 4, from which we could directly inspect the single-label classification and binary segmentation sub-tasks of each proposal embedding. Overall, the predicted classes are precise and confident, and the produced masks of proposal embeddings completely cover the corresponding semantic classes. Although *Truck* is actually not in the first example, the class score and binary mask are both relatively lower, and thus the fused result will not severely degrade the final segmentation performance.

Pixel-pixel Similarity Transfer. As shown in the 2nd row of Tab. 2, we solely enable pixel-pixel similarity transfer on the foundation, which leads to dramatical improvement. The in-depth quantitative analysis about the transferability is shown in Tab. 3. We collect base pixel pairs and novel pixel pairs sampled from 100 training image pairs, and employ GT labels to evaluate the similarities predicted by our SimNet. From the table, we can find that the semantic similarity learned from base classes is highly transferrable across classes. The in-depth qualitative visualization is given in Fig. 5, where we can see that for each sampled pixel, SimNet could "activate" the pixels belonging to the same semantic class across images. Therefore, the transferred pixel-pixel similarity could provide effective semantic relationship for distilling to regularize novel classes. The Fig. 6 (a) shows the impact of α (controlling the strength of distillation) to our model. Specifically, we change α in a range while fixing the other hyper-parameters, and depict the results obtained by our full-fledged model. Thus, our method is relatively robust to α in an appropriate range. Other experimental analyses including reference image selection and point number selection are left to Supplementary.

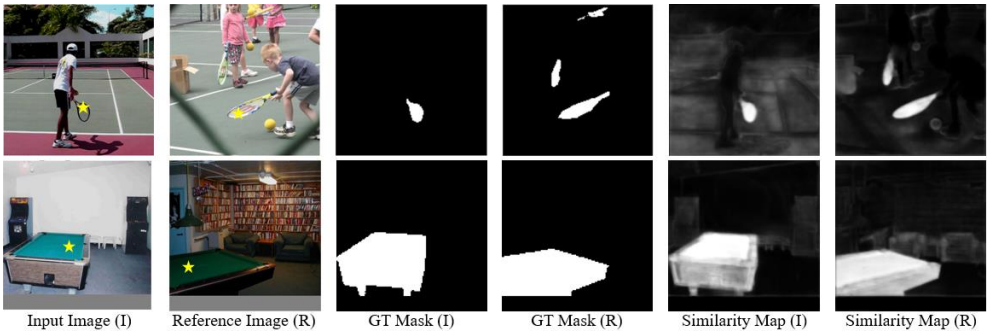


Figure 5: Visualizations for the pixel-pixel similarity on two datasets. In each row, the left two columns show the input image (I) and the reference image (R), where the yellow stars indicate the sampled pixels. The middle two columns depict the GT masks of the corresponding novel class. In the right two columns, we use SimNet to estimate the similarities between the sampled pixel in I (*resp.*, R) and all pixels in R (*resp.*, I), and get the similarity map (R) (*resp.*, (I)).

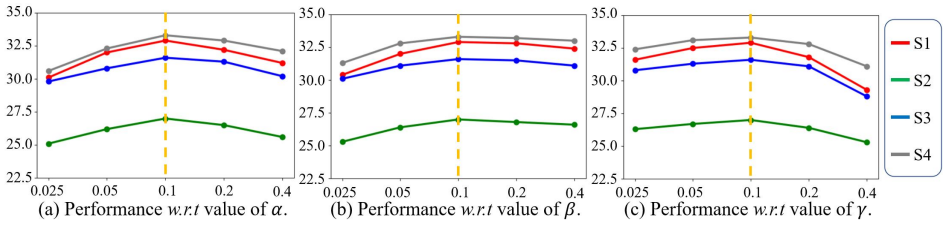


Figure 6: The performances of our model *w.r.t* different values of α, β, and γ on four splits of COCO-Stuff-10K dataset. The dotted lines indicate the default values.

Complementary Loss. As shown in the 3rd row of Tab. 2, we solely enable our complementary loss on the foundation, which also results in obvious promotion. In the 4th row of Tab. 2, with all the modules enabled, our full-fledged model achieves the optimal performance. The Fig. 6 (b,c) illustrate the impact of β (controlling the strength of injecting complementary insight) and γ (the prior probability for the ignore class) to our model. Larger γ will block the supervision to novel masks and degrade the performance. Smaller γ will "over-activate" novel classes on ignore pixels, which has relatively less impact. Overall, our model is robust to β and γ in an appropriate range.

4.4 Generalization Ability for Dataset Expansion

Focusing on a problem that uses cheaper labels to expand the off-the-shelf dataset, we naturally wonder the generalization ability of our method in such expansion. We investigate by further manipulating the datasets (*e.g.*, considering more novel classes). The experiments are left to Supplementary.

4.5 Limitation Discussion and Significance Test

We discuss the limitation and provide failure cases to further analyse our method. We also show that our method is statistically significant by significance test. The two sections are left to Supplementary.

5 Conclusion

In this paper, we focus on the problem of weak-shot semantic segmentation, which makes full use of the off-the-shelf pixel-level labels of base classes to support segmenting novel classes with image-level labels. For such problem, we have proposed a dual similarity transfer framework, which is built upon MaskFormer for performing proposal-pixel similarity transfer. To further regularize the model with pixel-level semantic relationship, we learn pixel-pixel similarity from base classes and distill such class-invariant similarity to novel masks. We also have proposed a complementary loss for facilitating the mask learning of novel classes. Extensive experiments on the challenging COCO-Stuff-10K and ADE20K datasets have demonstrated the effectiveness of our proposed method.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018.

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018.

[3] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via sub-category exploration. In *CVPR*, 2020.

[4] Junjie Chen, Li Niu, Liu Liu, and Liqing Zhang. Weak-shot fine-grained classification via similarity transfer. *NeurIPS*, 2021.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[6] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: exploring simple meta-learning for few-shot learning. In *ICCV*, 2021.

[7] Zitian Chen, Zhiqiang Shen, Jiahui Yu, and Erik Learned-Miller. Cross-supervised object detection. *arXiv preprint arXiv:2006.15056*, 2020.

[8] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 2021.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database.

[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[11] Ruochen Fan, Qibin Hou, Ming-Ming Cheng, Gang Yu, Ralph R Martin, and Shi-Min Hu. Associating inter-image salient instances for weakly supervised semantic segmentation. In *ECCV*, 2018.

[12] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. In *ACM MM*, 2020.

[13] Yuchen Guo, Guiguang Ding, Yue Gao, and Jungong Han. Active learning with cross-class similarity transfer. In *AAAI*, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. *ICLR*, 2017.

[16] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, 2018.

[17] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016.

[18] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *ICCV*, 2019.

[19] Suha Kwak, Seunghoon Hong, and Bohyung Han. Weakly supervised semantic segmentation using superpixel pooling network. In *AAAI*, 2017.

[20] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. *NeurIPS*, 2021.

[21] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *CVPR*, 2019.

[22] Yan Li, Junge Zhang, Kaiqi Huang, and Jianguo Zhang. Mixed supervised object detection with robust objectness transfer. *IEEE transactions on pattern analysis and machine intelligence*, 41(3):639–653, 2018.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[25] Yan Liu, Zhijie Zhang, Li Niu, Junjie Chen, and Liqing Zhang. Mixed supervised object detection by transferring mask prior and semantic similarity. In *NeurIPS*, 2021.

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[27] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016.

[28] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *BMVC*, 2017.

[29] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017.

[30] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 2017.

[31] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*, 2020.

[32] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

[33] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *ICCV*, 2021.

[34] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *ICCV*, 2021.

[35] Xiang Wang, Shaodi You, Xi Li, and Huimin Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *CVPR*, 2018.

[36] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020.

[37] Chuanguang Yang, Helong Zhou, Zhulin An, Xue Jiang, Yongjun Xu, and Qian Zhang. Cross-image relational knowledge distillation for semantic segmentation. *CVPR*, 2022.

[38] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. *NeurIPS*, 2020.

[39] Fei Zhang, Chaochen Gu, Chenyue Zhang, and Yuchao Dai. Complementary patch for weakly supervised semantic segmentation. In *ICCV*, 2021.

[40] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018.

[41] Tianyi Zhang, Guosheng Lin, Weide Liu, Jianfei Cai, and Alex Kot. Splitting vs. merging: Mining object regions with discrepancy and intersection loss for weakly supervised semantic segmentation. In *ECCV*, 2020.

[42] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[43] Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang. Boosting weakly supervised object detection with progressive knowledge transfer. In *ECCV*, 2020.

[44] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[45] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

[46] Siyuan Zhou, Li Niu, Jianlou Si, Chen Qian, and Liqing Zhang. Weak-shot semantic segmentation by transferring semantic affinity and boundary. *arXiv preprint arXiv:2110.01519*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 4.5.
   (c) Did you discuss any potential negative societal impacts of your work? [No] The focused problem is similar to the wide works of weakly supervised semantic segmentation, and has no obvious negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code and the data are proprietary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 4.5.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.1.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We use several public datasets and cite the original papers.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]